StegRithm: Steganographic Algorithm for Digital ASCII Text Documents

K. F. Rafat and M. Sher

***Abstract*—Information can not be measured in length and breadth but when received via some media, either in clear or encrypted form, can be assigned confidence level as 'authentic' (genuine) or 'misleading' (tempered with) because there always exist an equally likely probability for it to get detected, fabricated and or destroyed by an adversary during the course of transmission. Steganography - often referred to as art and / or science of hiding information together with Cryptography, is a way out for the aforementioned problem.**

Over the years we have come across several steganographic techniques / algorithms designed for hiding information inside digital media like Image, Audio, and Video files etc. This paper, however, is an attempt to propose a new steganographic algorithm (added with cryptography) for hiding information inside digital ASCII Text documents - an area of study regarded as 'difficult' by many because changing a single bit in an ASCII code of a text document may render it as erroneous character / word.

***Index Terms*—Covert channel, steganography, steganology, stealth communique, information hiding.**

I. Introduction

It may not be incorrect to tag the era we are living in as an era of Information Technology (IT) where gigantic advancement has been made in Computer and Telecommunication fields to an extent that Internet (a net of networks) has gained acceptance of being the preferred choice of communication, among masses and the Governments alike, and has also become a single efficient and cost-effective point of reference when it comes to information sharing.

Closely tied with this advancement are issues concerning confidentiality and integrity of the shared / exchanged information for which a variety of solutions have been proposed. However, among the solutions so proposed Cryptography and Steganography have gained world wide acclamation.

Cryptography and Steganography, with roots in Greek, are different in their approach towards information security, but yet seems able to provide near to perfect blended security solutions when it comes to protecting valuable information. Cryptography (or cryptology; from Greek κρυπτός, kryptos, "hidden, secret"; and ράφω, gráphō, "I write", or -λογία, -logia, respectively) [1], makes the information unintelligible whereas Steganography (from Greek words steganos (στεγανός) meaning "covered", and graphein (γράφειν) meaning "writing") [2], ensures that no clue or foot print of an information should exist.

ASCII, acronym for American Standard Code for Information Interchange, developed in 1963 with the intent that different computers can communicate with each other, is a subset of 256 character coding scheme, constituting 128 (0 ~ 127) codes i.e., seven bit code - 27, to represent the English character set. Of these 128 codes, first 32 characters are referred to as non-printing control characters including carriage return, line feed and bell (character 7) etc.

This paper is organized as follows: Section 2 discusses the model, terminology, approach, categorization and evaluation criteria for Steganographic System. Section 3 takes a look at relevant research work in the area of Text Steganography in context of our proposed algorithm whereas in Section 4 we elaborate on the problem domain. In Section 5 we present our proposed new Steganographic algorithm. Section 6 highlights the advantages and limitations of the proposed solution. Section 7 suggests future work to be done while Section 8 concludes the discussion.

II. Model, Terminology, Approach and Evaluation Criteria for Steganographic System

A. Model

Simmons' "Prisoners' Problem" (1984) [3] was the first to propose a model for secure (innocent) communication. He discussed the scenario as: Alice and Bob being locked up in prison where they are kept in separate cells, far at distance from each other, and want to decide on escape plan. They only way they can communicate is via plain (un encrypted) messages and if Eve detects that it is likely that she will thwart their efforts by transferring them to intense-security cells from where they can not communicate. Knowing this situation in advance, Alice and Bob agreed on a secret scheme, so that while in prison they can plan their escape by using this scheme in their messages without getting noticed by Eve.

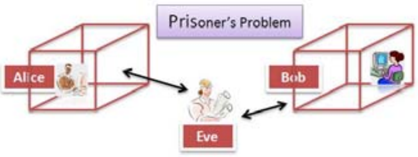


Fig. 1. Simon's security model

In April 1998, J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, Piotraschke, A. Westfeld, G. Wicke, and G. Wolf, proposed in [4] a improved security model for Steganographic system, where at a later point, they elaborated that the attacker may know *Cover* and *embedding*

Manuscript received September 20, 2012; revised November 3, 2012.

The authors are with International Islamic University, Islamabad, Pakistan (e-mail: farhan.phdcs35@ iiu.edu.pk).

DOI: 10.7763/IJET.2012.V4.480

process i.e., the Algorithm, but still he/she can not extract hidden information if the stego key is not known (Kerckhoff's Principle).

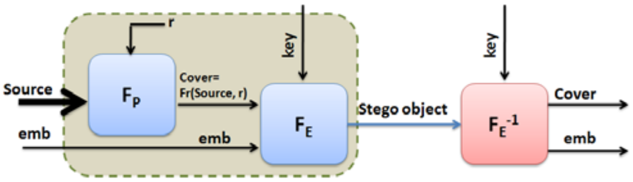


Fig. 2. Steganographic model with pre-processing

B. Terminology

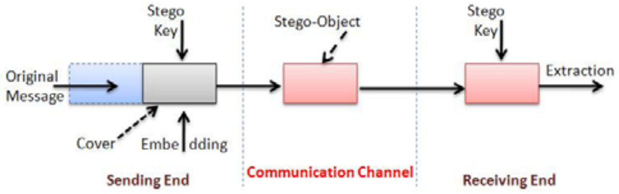


Fig. 3. Terminology used in steganography

The common terminology used with reference to Steganography explained in [5] includes:

- **Message**_Contents required to be hidden.
- **Cover / Carrier**_Actual container for hiding message.
- **Stego Object**_Resultant Cover carrying hidden message.
- **Stego Key**_A point of reference for hiding / un-hiding process.
- **Embedding**_Process for hiding information in Cover.
- **Extraction**_Process for retrieving hidden information from the Stego Object.

When you submit your final version, after your paper has been accepted, prepare it in two-column format, including figures and tables.

C. Approach

Steganography has been evolved from pure to private and then to public key in context of level of security associated with a Steganographic System as explained in [6]:

- **Pure Steganography**_The information embedding and extraction process does not operate under control of Stego Key i.e., anyone knowing the information embedding algorithm can reverse engineer the process to extract hidden message. [No Security – used prior to the introduction of Symmetric key concepts]
- **Private Key Steganography**_The process of message embedding and extraction operates under control of a Symmetric Key (same key is fed to the Steganographic System at the sending and receiving end on a pre-agreed key schedule). [Moderate Security Level]
- **Public Key Steganography**_Message embedding and extraction are performed using different Stego Keys. [Highest Security Level]

D. Categorization of Steganographic Systems

The authors in [6] have categorized the Steganographic Systems such as under:

- **Cover Generators**_Here the Steganographic System itself generates a Carrier / Cover on the basis of information to be hidden e.g., spam e-mailers etc.
- **Distortion Techniques**_The embedding is done by distorting the original signal and at the receiving end, the deviation of the received signal is measured for extracting information.
- **Spread Spectrum Techniques**_On the analogy of the technique used in Telecommunication sector where signal is distributed over a range of frequencies (bandwidth).
- **Statistical Methods**_Here statistical properties of the Carrier / Cover are altered and the receiver conducts hypothetical testing on the cover to retrieve information.
- **Substitution Systems**_Substitutes the part of Cover (usually the redundant one) with secret information.

E. Evaluation Criteria

Jonathan Cummins, Patrick Diskin, Samuel Lau and Robert Parlett at [7] have given the evaluation criteria for hiding information digitally.

- **Integrity**_The embedding process must be error free.
- **Perceptibility**_The resultant cover (stego object) shall retain its originality as close as possible.
- **Robustness**_Modifying stego object must not affect the embedded information (i.e., Watermark) [8].
- **Assumption**_The attacker knows that the '*carrier*' contains hidden information.

In 2010, the author at [9] has associated new attribute to the above criterion by suggesting that Stego ***key-entropy*** and ***key-length*** must also be taken into consideration at the time of evaluation.

III. Related Work

This section briefly discusses some of the known steganographic techniques in context of our proposed algorithm. However, for a study on latest text-based Steganographic techniques [10] is referred.

A. Hiding Data within White Spaces

In their paper at [11] the authors used white spaces in HTML cover file for hiding one byte of information per white space. The byte to be hidden is coloured by the same colour as of background colour of HTML page i.e., Secret Byte Colour = Web Page's Background Colour.

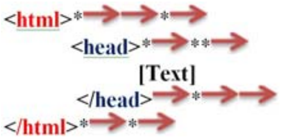
Limitation: Attacker knowing the algorithm can extract the hidden information.

B. Hiding Data in HTML Document

Author's in [12] elaborate that 'Steganos for Windows' a software program uses separators i.e. space and horizontal tab at the end of line of HTML document to hide binary bits ('1' and '0') of secret message.

Example:

If * denotes a *Space* and → the *horizontal-Tab* then:



i.e., bits 100101011001001010 are embedded inside the web page.

Limitations:

1) Increased Stego Object size.
2) Attacker knowing the algorithm can extract the hidden information.

In later discussion, the author showed use of line shift (0xA0 and 0xD0 HEX) in Windows and (0xA0 HEX) in Unix Operating System to interpret these as either ‘1’ or ‘0’. Majority of text editors can translate the two codes for line shift without ambiguity; hence is a secure way for hiding information.

C. White Spaces

One of the techniques suggested by W. Bender, D. Gruhl, N. Morimoto, and A. Lu in [13] is to places one or two spaces after every terminated sentence in the cover/text file to hide secret bit ‘0’ or ‘1’ as the case may be.

T	H	E		Q	U	I	C	K		B	R	O	W	N		F	O	X
J	U	M	P	S		O	V	E	R		T	H	E		L	A	Z	Y
D	O	G	.															

Fig. 4. Message

T	H	E		Q	U	I	C	K		B	R	O	W	N		F	O	X		
J	U	M	P	S		O	V	E	R		T	H	E		L	A	Z	Y		
D	O	G	.																	

Fig. 5. Stego object

IV. Problem Domain

The distinct characteristic of ASCII Text Documents is that these are written, saved and retrieved in the same manner as these appear before human eye, unlike other file formats like Images, Audio, Video etc. which are stored, saved and retrieved in different manner. Second, files other than ASCII Text files have additional information (Meta data) associated with them to facilitate the requirements associated with each file format. Third, less text files the other referred multimedia file formats exploits the limitations of human audio and visionary system.

V. Proposed Algorithm

A. Objectives

We want our algorithm to be as close to evaluation criteria as possible added with:

- Capable of hiding at least twice of information as compared to those discussed in Section III.
- Retention of Carrier/Cover files size by the Stego Object.
- The Stego Object to be easily converted into Portable Document Format (pdf) and be saved as MS Word File Format and vice versa without loosing the hidden information.

B. Techniques Involved

- **Substitution**_With reference to our defined objective in sub para above, we concentrated on finding as many character codes (in work space of 256) that can be *substituted* for ASCII character code 32 i.e., Space.
- **Symmetric Key Steganography**_We want the algorithm to be efficient in terms of time. We assume that the Sender and Receiver in stealth communication have pre-agreed on Stego Key usage and Stego Key Management System (SKMS).

C. Stego Key Length

The algorithm operates on a 256-bit key.

D. Design Basis

For our technique to work, we analyzed the UTF-8 coding scheme, being backward compatible with ASCII coding to identify characters that may be substituted for the blank ASCII character, for the sole purpose of information hiding, as Null (0), Tab (9), chr(160) & chr(255).

E. Implementation Aspect

Since we need to detect the behavior of digital Text Documents upon substituting blank character with those stated above, and also to judge the practicality of scheme in parallel, we embed the four alternates as substitutes for spaces in ASCII Text document, one by one, using Visual Basic 6 as tool and found out that we need to drop the Character code 255 while the reaming three codes together with ‘Space’ character serves the desired purpose.

F. Increased Embedding Capacity

At this point, we had four characters (in work space of 256) including ‘Space’, so we assigned a unique binary bit pair to each of these characters as under, to double the bit embedding capacity of our algorithm and for convenience we shall refer to these as *Steg Characters* in later discussion:

TABLE I: Character Codes with Binary Bit Pairs

		1	**2**	**3**	**4**
1	**Binary Bit Pair**	**00**	**01**	**10**	**11**
2	**Character**	**Null**	**Chr(9)**	**Space**	**Chr(160)**

However, to make the above assignment dynamic for security reasons, we decided to make this assignment key dependent as follows: Since there are four characters involved, hence their possible unique arrangements [14] are 12, and so we constructed a Look-Up table of 4 x 12 (random arrangements) and assigned a unique index value (1 ~ 12) to each of the 12 rows as under:

TABLE II: Look-up Table for Character Codes

Index	Col - 1	Col - 2	Col - 3	Col - 4
R-1	Null	Space	Chr(9)	Chr(160)
R-2	Chr(160)	Null	Chr(9)	Space
R-3	Chr(9)	Space	Chr(160)	Null
R-4	Space	Chr(160)	Null	Chr(9)
R-5	Chr(9)	Null	Space	Chr(160)
R-6	Chr(160)	Chr(9)	Space	Null
R-7	Space	Null	Chr(9)	Chr(160)
R-8	Null	Chr(160)	Space	Chr(9)
R-9	Space	Chr(9)	Chr(160)	Null
R-10	Chr(9)	Chr(160)	Null	Space
R-11	Null	Chr(9)	Chr(160)	Space
R-12	Chr(160)	Space	Null	Chr(9)

We added the values of 2nd, 5th and 8th Stego key bytes (any other combination can be used) and reduced the result to modulo 12 + 1, to have a random row number i.e., R-*x* where *x* is any number in range of 1 ~ 12. The permutation against R-*x* in Table – 2 will now serve as initializer for Table – I i.e., if R-*x* = 9 then Table – I will appear as shown below:

TABLE III: TABLE – I INITIALIZED

		1	2	3	4
1	**Binary Bit Pair**	**00**	**01**	**10**	**11**
2	**Character**	**Space**	**Chr(9)**	**Chr(160)**	**Null**

We further decided to reinitialize / rearrange these four characters after embedding any binary bit pair of secret information, to break any static relationship that might exist between message and these four Characters for which we referred to 256 bit Stego key. Here after embedding first binary bit pair of secret information inside the container (i.e., substituting any of the four Characters in Table-I with the first space in Cover text document) we set the Stego Key Byte counter (SK_BC) to the value of the first Stego Key byte and reduced it to modulo 12 + 1 after adding previous(SK_BC) to it e.g., Let for understanding assume that 1st five Stego Key values are 133, 27, 256, 11 & 192, then Table – III will take the form as given below:

TABLE IV: DYNAMIC INITIALIZATION

		1	2	3	4
SK_BC	Binary Bit Pair	00	01	10	11
	Character Code	Space	9	160	Null
SK_BC= Mod(133+0, 12)+1=2		Chr(160)	Null	Chr(9)	Space
SK_BC= Mod(27+133, 12)+1=5		Chr(9)	Null	Space	Chr(160)
SK_BC= Mod(256+27, 12)+1=8		Null	Chr(160)	Space	Chr(9)
SK_BC= Mod(11+256, 12)+1=4		Space	Chr(160)	Null	Chr(9)
SK_BC= Mod(192+11, 2)+1=12		Chr(160)	Space	Null	Chr(9)
…		…	…	…	…
Till end of embedding process					

G. Applying Encryption

To make the task of extracting message out of embedded bits from Stego Object cumbersome, we first performed modulo 2 addition of message bits with Stego Key bits and then reused the resultant (ciphered) bits as Stego Key bits for their subsequent addition with next message bits till end of that message i.e., Cipher feed back (CFB) mode.

H. Message Header

Sender first constructs a message header of 7 bytes whose distribution is given as under and then appends it before the actual message:

TABLE V: MESSAGE HEADER

Length of Message file	Message file extension
4 bytes	4 bytes

I. Bit Embedding (At Sending End)

Following steps are performed for hiding message inside a Text Carrier / Cover file:

i. Select the pre-agreed Stego key and initialize Table-I.
ii. Type / Select a message (message form can vary from Image, Text, Audio to Video file etc).
iii. Perform modulo 2 additions of message and Stego Key bits.
iv. Reuse resultant bits from step (iii) as Stego Key bits if message exceed Stego Key length. Repeat step (iii).
v. Ciphered bits of step (iii) be grouped into pairs of two bits each.
vi. Start from 1st pair of bits of step (v) and locate corresponding Character written against it in Table–III.
vii. Locate 1st occurrence of ‘Space’ in the text file.
viii. If the corresponding Character of step (vi) is not space then substitute ‘Space’ in step (vii) with that character otherwise leave that ‘Space’ at its location.
ix. Re-initialize *Steg Characters*.
x. Take the next pair of grouped bits of step (v) and locate corresponding Character against it in Table–III.
xi. Locate next occurrence of ‘Space’ in the text file.
xii. If the corresponding Character of step (x) is not space then substitute ‘Space’ in step (xi) with that character otherwise leave that ‘Space’ at its location.
xiii. Repeat steps (ix) to (xii) till the end of the two bit ciphered groups obtained from step (v).

J. Bit Extraction (At Receiving End)

To retrieve the embedded information from Carrier / Cover Text file, following steps are to be performed:

i. Select the pre-agreed Stego key and initialize Table-I.
ii. Find the 1st *Stego Character* in the Carrier file, check its code value as 32, 9, 160 or 0 and store the binary bit pair written against it in Table – III, for later use.
iii. Re-initialize Table – III.
iv. Find next *Stego Character* in the Carrier file, check its code value as 32, 9, 160 or 0 and concatenate the binary bit pair written against it as in Table – IV with that obtained in step.
v. Repeat steps (iii) and (iv) until all *Stego Characters* are translated into binary bit pairs.
vi. Perform modulo 2 addition of the bit string obtained from step (v) with Stego Key bits.
vii. Reuse the resultant bits from step (vi) as Stego Key bits if bit string of step (v) exceed Stego Key length.
viii. Divide the string of binary bits of step (vii) into chunks of eight bits each and, translate these into equivalent UTF-8 bit code till the message header is obtained. From here we will get the length of embedded file and its type i.e., extension.
ix. Now repeat steps (vi) to (vii) till the end of message.

Divide the string of bits into chunks of eight bits each and write its equivalent UTF-8 code in file (having being assigned the extension obtained in step (viii) for later retrieval.

VI. ADVANTAGES AND LIMITATION

A. Advantages

1) Absolutely Stego Key dependent while embedding and / or extracting hidden bits hence adheres to Krickhoff’s Principle.
2) Hides bits of secret information, hence any file types can now be hidden inside ASCII Text documents.

3) Capable of hiding twice the information than any of its predecessor data hiding schemes.
4) Each *Stego Character* can have any of the four bit pairs i.e., 00, 01, 10 or 11, thereby making the intruders task of reverse engineering the process cumbersome.
5) Length of Stego Object remains unchanged.
6) Saving the Stego Object in 'pdf' and 'doc' file formats retains the embedded information.
7) Re -saving the Stego Object (saved earlier in 'doc' format) will not lose the embedded information.
8) Encryption prevents un-authorized users from translating the extracted bits of information into message.
9) Message header (together with embedding algorithm) guarantees fulfillment of evaluation criteria as under:
   - Robustness.
   - Retains message integrity.
   - Correct extraction of hidden bits.
   - File extension helps in identifying the type of embedded information and for its automatic opening / display when gets implemented.

B. Limitation

Perceptability_Frequent occurrence, if any, of 'Tab-Chr(9)' used as *Stego Character* in our proposed algorithm, may cause inconvenience to user on receipt of Stego text Object.

VII. Future Work

The proposed algorithm has been checked manually and the validity of *Stego Characters* is confirmed by embedding these in ASCII Text document files using Visual basic 6 as tool and following are suggested as future work:

1) Software Implementation.
2) Analysis in terms of speed and memory usage.
3) Compressing messages before encryption using appropriate compression algorithms supporting / evolved for specific message file format.

VIII. Conclusion

This paper introduces a novell, one of a kind Stegnaographic algorithm, capable of embedding twice much information in a Carrier / Cover Text file then any of its counterparts. Use of encrypion just before embedding also distinguishes it from the rest of the text-based steganographic techniques in use for the purpose. Message header tells the receiver in advance about the length and type of information being stored in the Carrier / Cover, there by increasing efficiency and performance.

References

[1] D. E. Knuth, "The Art of Computer Programming," vol. 2, *Semi numerical Algorithms*, ed.3 pp.145-146.

[2] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

[3] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology: Proceedings of Crypto 83* (D. Chaum, ed.), pp. 51–67, Plenum Press, 1984.

[4] J. Zollner, H. Federrath, H. Klimant, A. Pritzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf. "Modeling the security of steganographic systems," in *2nd International Workshop Information Hiding*. Springer, Berlin/Heidelberg, German, vol. 1525: 345 - 255, 1998.

[5] B. Pfitzmann, "Information hiding terminology," in *Information Hiding, First International Workshop* (R. Anderson, ed.), vol. 1174 of Lecture Notes in Computer Science, pp. 347–350, Springer, 1996.

[6] D. Kleiman, K. Cardwell, T. Clinton, M. Cross, M. Gregg, and J. Varsalone, *The Official CHFI Study Guide (Exam 312-49) for Computer Hacking Forensic Investigators,* Published by: Syngress Publishing, Inc., Elsevier, Inc., 30 Corporate Drive, Burlington, MA 01803, Craig Wright.

[7] P. Diskin, S. Lau, and R. Parlett, "Steganography and Digital Watermarking," Jonathan Cummins, School of Computer Science, The University of Birmingham, 2004.

[8] S. Channalli and A. Jadhav, in their research paper titled, "Steganography An Art of Hiding Data," in *International Journal on Computer Science and Engineering,* vol.1(3), 2009, pp.141.

[9] K. F. Rafat and M. Sher, "IST: Improved Steganography for HTML," *9th European Conference on Information Warfare and Security*, Hosted by strategyinternational.org and the Department of Applied Informatics University of Macedonia Thessaloniki, Greece 1-2 July 2010, pp. 366-376

[10] K. F. Rafat and M. Sher, "Survey report – State of the art in Digital Steganography focusing ASCII Text Documents," *International Journal of Computer Science and Information Security (IJCSIS),* vol. 7, no. 2, 2010, ISSN 1947-5500.

[11] Ala"a H. Al-Hamami, Mazin S. Al-Hakeem, and Mohammed A. Al-Hamami. (November 9, 2009). A Proposed Method to Hide Text Inside HTML Web Page File. [Online]. Available: http://mohammad.iraq.ir/A%20prposed%20method%20to%20hide%20text%20inside%20html%20web%20page%20file.pdf

[12] Stanislav S. Barilnik, Igor V. Minin, and Oleg V. Minin , "Adaptation of Text Steganographic Algorithm for HTML," Novosibirsk State Technical University.

[13] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, issues 3&4, pp. 313-336, 1996.

[14] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

Prof. Dr. M. Sher is Chairman of Department of Computer Science, Faculty of Basic and Applied Sciences, International Islamic University, Islamabad. He received B.Sc. degree from Islamia University Bahawalpur and M.Sc. degree from Quaid-e-Azam University, Islamabad, Pakistan. He received Ph.D. degree from TU Berlin, Germany in Computer Science and Electrical Engineering. His area of research is Next Generation Networks Security. He has 22 years research & development and teaching experience in Pakistan & Germany.

Dr Sher supervised more than 26 BSCS/MCS projects, 22 MSCS Thesis and 4 PhD Thesis in the area of Information Security and Computer Networks. At present he is supervising 8 PhD and 9 MS students. He has more than 60 publications in International peer reviewed conferences and International journals.

He received National I.T. Excellence Award & Gold Medal from NCR in the field of Research & Development for the period of 1997-2000 on his research work. He was given Best Teacher Award by Higher Education Communication (HEC) of Pakistan in 2008. In 2011 he has been awarded "Senior Information Security Professional of Asia 2011" by International Information System Security Consortium (ISC)^2 Asian Advisory Board under Asia-Pacific Information Security Leadership Achievements (ISLA) Awards Program. He also received 10th Teradata National IT Excellence Award in the category of "Excellence in IT Education" on 17th March 2012.

He is member of National Curricula Committees of the Higher Education Communication for Computer Science and Information Technology. He is also leading "Urdu Internet Security Research Working Group (UISR-WG)" of Urdu Internet Society and is an Active Member of IGFPAK.

Khan Farhan Rafat is a PhD Scholar at International Islamic University, Islamabad, Pakistan. He did Masters in Computer Science in 2004 from Gomal University, D.I.K. followed by MS in Telecommunication Engineering in 2007 from University of Management and Technology (UMT), Lahore, Pakistan.